0. Download the Starter Kit from onCourse. Make a new C++ Project. Be prepared to participate in a discussion.

1. A "**canonical object**" in C++ (at a minimum) contains the following methods:

2. Draw the inheritance diagram of the code in this project below:

3. In `main()`:
   - (a) Create a `Faculty` object (with the name, **`F`**) on the runtime stack called "Pete", 22 years of age without tenure.
   - (b) When creating a `Faculty` object, what is the order of (parent, child) CTORs called?
   - (c) Have Pete greet Mary.
   - (d) Have Pete give his farewell message.

4. In `main()`:
   - (e) Using a pointer to an `Employee` (parent object) called **`pE`**, create a new `Faculty` object on the heap called "Mark", 44 years of age who does have tenure.
   - (f) Have Mark greet Sal.
   - (g) Have Mark give his farewell message.
   - (h) Remove the `virtual` keyword from `Employee::greeting()`. Rerun the code. What happens? Explain why.
   - (i) Put the virtual keyword back.

5. Declare an Employee object on the runtime stack. What happens? Why? What could you do to "fix" this?

6. Pass Pete's object (`F`) *by value* to the testCopyCTOR() function. Explain the control of flow from `main()` to the CTORs to the function to the DTORs.

7. Make a new copy (in `G`) of the object (`F`) at the time of `G`'s declaration as shown below. Explain what happens?

```
Faculty G(F);
```

8. Delete Mark's (dynamically allocated) object from the heap.

9. Describe the calling of the DTORs at the end of your program for a given object.

10. **Practice Polymorphism**. Declare an array of 10 pointers to Employee objects. Make two new statically declared Faculty objects ("Jane" and "Tarzan"). Assign the addresses of these two objects and your other Faculty objects already statically declared into the first four slots of your array of pointers.

    Write a loop to iterate over your array of pointers, each time having the pointer give the greeting of the respective object.

11. Make a copy of `Faculty.h` and `Faculty.cpp` and make a new class called `Student` that should also inherit from `Employee`. **Make the Student greeting be unique, e.g., "Dude! What's up!"** Make some statically declared `Student` objects and assign their addresses to your array of `Employee` pointers. Iterate over all these `Employee` pointers and notice the greetings.

12. Make another new (statically declared) object (say, H) and assign F to H. You will have to write the definition of the **`operator=`** method, of course.